4/5/1 (Item 1 from file: 347)

04233284 **Image available**
DEBUG BACK-UP DEVICE FOR DECENTRALIZED PROGRAM

PUB. NO.: 05-224984 **JP 5224984** A]
PUBLISHED: September 03, 1993 (19930903)
INVENTOR(s): ARAGAKI NORIKO
YAMAZAKI KENICHI
TENKAI RYOJI
APPLICANT(s): NIPPON TELEGR & TELEPH CORP <NTT> [000422] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 04-025405 [JP 9225405]
FILED: February 12, 1992 (19920212)
INTL CLASS: [5] G06F-011/28; G06F-011/34; G06F-015/16
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.4 (INFORMATION PROCESSING -- Computer Applications)
JOURNAL: Section: P, Section No. 1659, Vol. 17, No. 677, Pg. 6, December 13, 1993 (19931213)

ABSTRACT

PURPOSE: To provide the subject back-up device which has the high universal applicability with no remodeling of the exclusive hardware and an operating system nor the change of 8 source program.

CONSTITUTION: A 1st device group 1 is built into each process and includes a means 3 which collects the execution history data on the processes and an execution history data transmitting means 4. A 2nd device group is built into a debugging host machine 10 connected to a decentralized processing system and includes a means 5 which receives and collects the execution history data from the group 1, an analyzing means 6 which rearranges the execution history data in the order of the causal relation and analyzes these data, and a means 7 which displays the analyzing result of the means 6.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平5－224984

(43)公開日　平成5年（1993）9月3日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　11/28		Ｊ　9290－5Ｂ		
11/34		Ｂ　9290－5Ｂ		
15/16	４５０	Ｄ　9190－5Ｌ		

審査請求　未請求　請求項の数１（全　５　頁）

(21)出願番号　特願平4－25405

(22)出願日　平成4年（1992）2月12日

(71)出願人　000004226
日本電信電話株式会社
東京都千代田区内幸町一丁目1番6号

(72)発明者　新垣　紀子
東京都千代田区内幸町1丁目1番6号　日本電信電話株式会社内

(72)発明者　山崎　憲一
東京都千代田区内幸町1丁目1番6号　日本電信電話株式会社内

(72)発明者　天海　良治
東京都千代田区内幸町1丁目1番6号　日本電信電話株式会社内

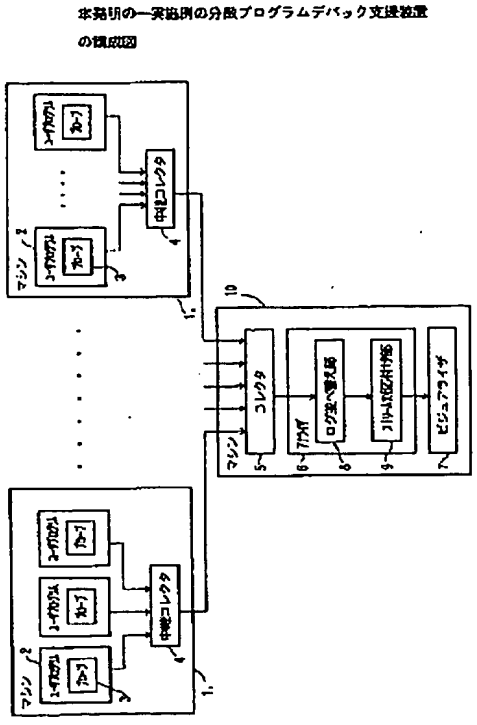
(74)代理人　弁理士　伊東　忠彦

(54)【発明の名称】分散プログラムデバッグ支援装置

(57)【要約】

【目的】　本発明の目的は、専用のハードウェア、オペレーティングシステムの改造やソースプログラムの変更を必要とせずに、汎用性の高い分散プログラムデバッグ支援装置を提供することである。

【構成】　本発明は、プロセスにそれぞれ組み込まれ、プロセスの実行履歴データを採取する手段３と、実行履歴データを送信する手段４を含む第１の装置群１と、分散処理システムに接続するデバッグ用ホストマシン１０に組み込まれ、第１の装置群１から送信された実行履歴データを受信し、収集する手段５と、因果関係のある順に実行履歴データを並べ替え、並び替えられた実行履歴データを解析する解析手段６と、解析手段６により得られた解析結果を表示する手段７を含む第２の装置群とを有する。



1

【特許請求の範囲】

【請求項１】　複数のプロセス間で通信を行い、ひとつの処理を行う分散処理システムにおいて、

前記プロセスにそれぞれ組み込まれ、前記プロセスの実行履歴データを採取する手段と、該実行履歴データを送信する手段を含む第１の装置群と、

前記分散処理システムに接続するデバッグ用ホストマシンに組み込まれ、前記第１の装置群から送信された前記実行履歴データを受信し、収集する手段と、因果関係のある順に前記実行履歴データを並べ替え、並び替えられた実行履歴データより送信イベントと受信イベントを対応させることにより前記プロセスの実行状況を解析する解析手段と、該解析手段により得られた解析結果を表示する手段を含む第２の装置群とを有することを特徴とする分散プログラムデバック支援装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、分散プログラムデバッグ支援装置に係り、特に、マルチプロセッサシステム及び、分散システム等の異なるプロセッサ上に存在するプログラム実行の主体である複数プロセス間で、通信を行いながら１つの処理を行う分散プログラムにおいて、各プロセスの実行状況を、実行順に実時間で表示することにより、分散プログラムのデバックを支援する分散プログラムデバッグ支援装置に関する。

【０００２】

【従来の技術】従来、分散プログラムをデバッグするための実行状況を表示する装置としては、オペレーティングシステムを改造して、プロセスが実行した実行履歴を採取するもの(MILLER, B. D., MACRANDER, C., AND SECHREST, S 1986. A distributed programs monitor for Berkeley UNIX. Softw. Pract. Exper. 16, 2, 183-200)、或いは、専用のハードウェアで実行履歴を採取するもの（RUBIN, R. V., RUDOLPH, L., AND ZERNIK, D. 1988. Debugging parallel programs in parallel. In Proceedings of Workshop on Parallel and Distributed Debugging. ACM. Published as SIGPLAN Notices 24, 1 (January 1989). pp. 216-225）、また、ユーザがソースプログラムを書替えるもの(SOCHA, D., BAILEY, M. L., AND NOTKIN, D. 1988. Voyeur: Graphical views of parallel programs. In Proceedings of Workshop on Parallel and Distributed Debugging. ACM. Published as SIGPLAN Notices 24, 1 (January 1989). pp. 206-215)が主体であった。さらに、分散プログラムで一般的によく使用される通信手段であるストリーム通信を行うプログラムをデバッグできない対話型分散デバッガ（HARTER, P. K., JR., HEIMBIGNER, D. M., AND KING, R. 1985. IDD: An interactive distributed debugger. In Proceedings of the 5th International Conference on Distributed Computing Systems. IEEE, pp. 498-506）等がある。

2

【０００３】

【発明が解決しようとする課題】しかしながら、上記従来のシステムは、専用のハードウェア上でしか行うことができないために、他のハードウェアとの互換性がない。また、ハードウェア毎に、オペレーションシステムの改造やユーザによるソースプログラムの変更が必要であり、汎用的ではないという問題がある。また、従来のシステムは、分散プログラムで一般的によく使用される通信手段であるストリーム通信を用いているプログアムをデバッグできないという問題がある。ストリーム通信の取扱いが困難であるのは、複数の送信によるデータを一回で受信するため、送信イベントと受信イベントが必ずしも１対１に対応しない可能性があり、送信プロセスのどの送信イベントに対応するのかの判断が難しくなるためである。因って、限られた分散プログムしかデバッグできない。

【０００４】本発明は上記の点に鑑みなされたもので、分散プログラムのデバッグにおいて、専用のハードウェア、オペレーティングシステムの改造や、ソースプログラムの変更等を必要とせずに汎用性の高い分散プログラムデバッグ支援装置を提供することを目的とする。

【０００５】

【課題を解決するための手段】本発明は、複数のプロセス間で通信を行い、ひとつの処理を行う分散処理システムにおいて、プロセスにそれぞれ組み込まれ、プロセスの実行履歴データを採取する手段と、実行履歴データを送信する手段を含む第１の装置群と、分散処理システムに接続するデバッグ用ホストマシンに組み込まれ、第１の装置群から送信された実行履歴データを受信し、収集する手段と、因果関係のある順に実行履歴データを並べ替え、並び替えられた実行履歴データより送信イベントと受信イベントを対応させることによって、プロセスの実行状況を解析する解析手段と、解析手段により得られた解析結果を表示する手段を含む第２の装置群とを有する。

【０００６】

【作用】本発明の分散プログラムデバッグ支援装置は、プロセス間通信を行っている分散プログラムから実行履歴データを採取する処理部分を、被デバッグプログラム（ユーザプログラム）に組み込むことにより、複数プロセスの実行状況や、動作、プロセス間の通信をリアルタイムで確認することができ、既存のハードウェアやオペレーティングシステムにも容易に対応する。また、実行履歴データ中の送信イベントと受信イベントを対応させることにより、ストリーム通信を扱っているプログラムのデバッグを支援できる。

【０００７】

【実施例】以下、本発明の実施例について図面とともに説明する。図１は、本発明の一実施例の分散プログラムデバッグ支援装置の構成図である。分散プログラムデバ

ック装置がリアルタイムで表示装置等にプロセスの実行状況等を表示するためには、実行履歴（以下ログと呼ぶ）データをプログラム実行時に、ユーザプログラムから採取し、１箇所のバッファプールに収集しておく必要がある。同図に示す複数のマシン1_1～1_nは、各々被デバッグ分散プログラムであるユーザプログラム２と、ログデータを作成するために、ユーザプログラム２に埋め込まれているプローブ３と、マシン１内のログデータを収集するための中継コレクタ４を有する。

【０００８】また、ホストマシン１０は、全マシン1_1～1_nより収集されたログデータをまとめて収集するコレクタ５、コレクタ５のログデータを解析するアナライザ６、アナライザ６の解析結果を表示するビジュアライザ７を有する。そのうち、アナライザ６は、ログデータを因果関係のある順に並び替えるログ並べ替え部８と、送信イベントと受信イベントの対応付けを行うストリーム対応付け部９を有する。

【０００９】次に、上記の各装置について説明する。ユーザ プログラム２に埋め込まれているプローブ３は、ユーザプログラム２からのログデータの採取を行い、中継コレクタ４へログデータの送信を行う。ここでは、対象とする分散プログラムをＣ言語としており、Ｃ言語のライブラリ関数の実行をイベントとして、ログデータを作成する。このログデータは、実行状況をリアルタイムでビジュアルライザ７に表示されるためのデータと、イベントの因果関係を表示するためのデータから構成されている。

【００１０】図２は本発明の一実施例のログデータを示す。ログデータは、受信イベントや送信イベント等のイベントの種類、プロセスＩＤ、イベント番号、プロセス内の実行時間、プロセスのポート番号、通信相手のポート番号、通信バイト数及び親プロセスＩＤ等である。

【００１１】実際にプローブ３が中継コレクタ４にログデータを送るタイミングは、ログデータを採取すべき関数が終了した時点である。終了した時点とするのは、その関数の終了状態を、ログデータに含めて送信するためである。なお、受信関数等の待ち時間を調べたい関数については、関数の実行の前にもログデータの送信を行う。

【００１２】中継コレクタ４は、各マシン１でユーザプログラム２から受け取ったログデータを一度蓄積し、一定数に達した時点、或いは、一定時間毎にホストマシン１０のコレクタ５に送信する。中継コレクタ４でまとめて送信することによって、ユーザプロセスから直接コレクタ５にログデータを送信するのに比べ、ネットワークを介する通信回数が減少するために、ネットワークへの負荷を減らす効果がある。

【００１３】ホストマシン１０のコレクタ５は、中継コレクタ４から収集したログデータをアナライザ６に送出する。アナライザ６は、到着順序の入れ替わったログデータを送信順に並べ替える部分であるログ並び替え部８と、ストリーム通信の対応付けを行う部分であるストリーム対応付け部分９から構成されており、ログ並び替え部８は、ログデータをランポートの論理時間（L. Lamport, Time, Clocks, and the Ordering of Events in a Distributed System, Communications of the ACM, Vol 21, No7(July 1978). pp. 558-565)などによって、因果関係のある順に並べ替える。これは、イベントの発生順にログデータが届かない可能性があるためである。ストリーム対応付け部９は、ログデータ並べ替え部８から並び替えられたログデータを受け取り、次の手順で送信イベントと受信イベントの対応付けを行う。なお、ここで、“対応付け”とは、表示画面等でイベント間を線で結ぶこと等を意味する。

【００１４】図３にストリームの対応付けを行うための表を示す。まず、このような表を各ストリーム毎に用意し、参照ポインタと、書込みポインタを０にする。ストリーム対応付け部分９は、“ストリームへ送信”というイベントのログデータを受け取ると、ログデータの内容に含まれている送信バイト数と、イベントＩＤ（図２）を図３の書込みポインタの指す欄に書込み、書込みポインタをインクリメントする。また、“ストリームからの受信”とういイベントを受け取ると、ログデータから受信バイト数を取り出し、図３の参照ポインタの指す欄の送信バイト数と、受信バイト数を比較する。

【００１５】ここで、参照ポインタがｉ欄を指しているとする。この時の送信バイト数をN_i、送信イベントＩＤをSE_iとし、受信バイト数をＭ、受信イベントＩＤをＲＥとする。比較結果は、次の３通りの場合がある。

①受信バイト数（Ｍ）より送信バイト数（Ｎｉ）が大きい時　（Ｍ＜Ｎｉ）：ｉ欄のイベントＩＤ（ＳＥｉ）と受信イベントＩＤ（ＲＥ）を対応させ、さらに、受信バイト数（Ｍ）と送信バイト数（Ｎｉ）の差をＮｉの代わりに、ｉ欄に書き込む。

②受信バイト数（Ｍ）より送信バイト数（Ｎｉ）が等しい時　（Ｍ＝Ｎｉ）：ｉ欄のイベントＩＤ（ＳＥｉ）と受信イベントＩＤ（ＲＥ）を対応させ、参照ポインタをインクリメントし、１＋ｉ欄に進める。

③受信バイト数（Ｍ）より送信バイト数（Ｎｉ）が小さい時　（Ｍ＞Ｎｉ）：ｉ欄のイベントＩＤ（ＳＥｉ）と受信イベントＩＤ（ＲＥ）を対応させ、参照ポインタをインクリメントし、１＋ｉ欄に進める。さらに、送信バイト数（Ｎｉ）と受信バイト数（Ｍ）の差と、次のｉ＋１の欄と比較し、上記の送信バイト数と受信バイト数を比較する操作を行う。

以上の操作により、１対１とは限らないストリーム通信の送信イベントと受信イベントの対応付けを行うことができる。

【００１６】ビジュアライザ７は、ログデータを因果関係のある実行順に（発生したイベント順）リアルタイム

5

で表示する。また、プロセス間通信のログデータは、アナライザ６で解析したデータをもとに表示する。

【００１７】上記のように本発明は、ビジュアライザ７により各プロセスの実行状況を実行順にリアルタイムで表示するので、時間軸に沿って、実行時に観察または、実行後に観察することができる。さらに、既存の分散プログラムに容易に組み込むことができるため、専用のハードウェアが不要となり、ユーザによるプログラムの変更も不要であるので、容易に分散プログラムのデバックを支援することができる。

【００１８】

【発明の効果】上述のように本発明の分散プログラムデバッグ支援装置によれば、ログデータを採取する機能をユーザプログラムに組み込むことによって、専用のハードウェア、オペレーティングシステムの改造を必要とせず、分散プログラムのデバッグにおいて、汎用性の高いデバッガ効果を得ることができる。また、今まで取り扱えなかったストリーム通信を取り扱えるため、より多くの分散プログラムのデバッグが可能になる。

6

【図面の簡単な説明】

【図１】本発明の一実施例の分散プログラムデバッグ支援装置の構成図である。

【図２】本発明の一実施例におけるログデータの内容を示す図である。

【図３】本発明の一実施例のストリーム通信の送信データと受信データの対応付けを行う際に、用いられるデータ構造を示す図である。

【符号の説明】

10　1_1～1_n　マシン
2　ユーザプログラム
3　プローブ
4　中継コレクタ
5　コレクタ
6　アナライザ
7　ビジュアライザ
8　ログ並べ替え部
9　ストリーム対応付け部
１０　ホストマシン

【図２】

本発明の一実施例におけるログデータの内容を示す図

イベントの種類
プロセスＩＤ
イベント番号
プロセス内の実行時間
プロセスのポート番号
通信相手のポート番号
通信バイト数（送信、受信）
親プロセスＩＤ

【図３】

本発明の一実施例のストリーム通信の送信データと受信データの対応づけを行なう際に用いられるデータ構造を示す図

送信バイト数	イベントＩＤ	
N_1	SE_1	
N_2	SE_2	
・・・	・・・	
N_i	SE_i	← 参照ポインタ
N_{i+1}	SE_{i+1}	
・・・	・・・	
N_1	SE_1	← 書き込みポインタ

【図１】

本発明の一実施例の分散プログラムデバック支援装置の構成図